IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 28 June 2022, accepted 14 July 2022, date of publication 19 July 2022, date of current version 29 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3192441

RESEARCH ARTICLE

A Cross Working Condition Multiscale Recursive Feature Fusion Method for Fault Diagnosis of Rolling Bearing in Multiple Working Conditions

ZHIQIANG ZHANG, FUNA ZHOU, AND SIJIE LI
School of Logistics Engineering, Shanghai Maritime University, Shanghai 201306, China
Corresponding authors: Funa Zhou (zhoufn2002@163.com) and Zhiqiang Zhang (zqiang_zhang@foxmail.com)

This work was supported by the National Natural Science Foundation of China under Grant 62073213.

ABSTRACT As a key component of electromechanical equipment in the intelligent manufacturing process, rolling bearings play an important role to secure a safe, stable, and efficient operation. Deep learning can be used to guide a data-driven fault diagnosis which requires that all data are independently identically distribution (i.i.d). When the equipment is operated with multiple working conditions, the collected samples violates the assumption of i.i.d, which will inevitably make it difficult to extract accurate feature involved in the data. This paper proposes a deep learning based fault diagnosis model to recursively fuse the multiscale feature on cross working conditions, such that data without working condition label can also be referred to train a satisfying deep learning model for fault diagnosis of bearing operated in multiple working conditions. In the case when only a small number size of training samples for a separated working condition are available, the proposes fusion mechanism aims to establish a jointly learning mechanism between different working conditions. To verify the effectiveness of the proposed algorithm, experimental validation was performed using the Case Western Reserve University (CWRU) rolling bearing public data set. The experimental results show that the proposed method can make full use of a small amount of labeled data with working conditions and a large amount of labeled data without working conditions. In ten types of fault diagnosis tasks with different fault sizes, the fault diagnosis accuracy reaches more than 94% for 4 working conditions and more than 86% for 8 working conditions.

INDEX TERMS Fault diagnosis, feature fusion, multiple working conditions, rolling bearing.

I. INTRODUCTION

With the increasing complexity of large electromechanical equipment, it has become increasingly difficult to perform intelligent fault diagnosis for them. Electric motors play an important role in the power system of electromechanical equipment and can cause catastrophic losses if they fail. Rolling bearings are one of the most critical components of electric motors, serving to support the rotating shaft and shaft components. Affected by factors such as overload, aging, and complex working conditions, rolling bearings are prone to failure. Therefore, the research on the intelligent fault diagnosis of rolling bearings is of great significance to ensure the safe and stable operation of electromechanical equipment and has received great attention from experts in the field of fault diagnosis [1]–[4].

Due to the complexity of industrial systems, it is difficult to establish an accurate physical model of them. On the other hand, the operation of the equipment generates a huge amount of monitoring data, and how to use these monitoring data to meet the high requirements of equipment safety and stability has likewise become an important issue worth studying. Data-driven fault diagnosis methods can build end-to-end fault diagnosis models using massive amounts of monitoring data, without being limited to precise physical models and expert knowledge information [5]–[8].

Deep learning is an effective tool for data feature extraction. Through a deep neural network structure, it can abstract

The associate editor coordinating the review of this manuscript and approving it for publication was Donato Impedovo.

data features to more precise feature scales layer by layer, thus extracting key features that reflect the characteristics of the data. Some researchers have performed fault diagnosis by combining deep learning methods with traditional signal processing methods [9]–[11]. Wang *et al.* [12] used wavelet packet analysis to decompose and reconstruct the original signal to reduce noise and then used support vector machines to classify the extracted features to improve the accuracy of fault diagnosis. Akhenia *et al.* [13] used methods such as short-time Fourier transform to generate two-dimensional time-frequency spectrograms from various fault conditions of bearings and then used deep learning algorithms as fault diagnosis classifiers to improve the fault diagnosis accuracy. Zhou *et al.* [14] used the overall empirical modal decomposition method to decompose the vibration signal into multiple intrinsic modal functions and then performed principal component analysis on them, and finally used LSTM networks for fault diagnosis to improve the accuracy of fault diagnosis.

The above researches aim to improve the accuracy of fault diagnosis by combining signal processing methods with deep learning methods, but it needs to pre-process the data using signal processing methods, which will affect the real-time performance of fault diagnosis. Therefore Zhou *et al.* [15] designed a sparse gate structure of Deep neural network (DNN) to reduce the propagation of useless information in the feature extraction process of neural networks and obtained better results in the fault diagnosis of rolling bearings. Hoang *et al.* [16] extracted features from data collected by multiple sensors using DNN, to improve the effectiveness in bearing fault diagnosis compared to the results from a single sensor.

The above researches aim to build relatively accurate fault diagnosis models by traditional deep learning methods, which do not require pre-processing of data and ensure the real-time fault diagnosis. However, it does not take into account the problem of multiple working conditions during the operation of the equipment. Multiple working conditions refer to the change of load during equipment operation which causes the change of motor speed, thus making the bearings work at different speeds. This problem of multiple working conditions of bearings destroys the consistency of data distribution and leads to difficulties in extracting features for fault diagnosis models.

Traditional fault diagnosis methods based on deep learning require a large number of labeled samples, but multiple working conditions lead to difficulties to label data, so the number of high-quality labeled data is small. If the working conditions are distinguished in advance and modeled separately, it does not guarantee real-time fault diagnosis, and the small amount of labeled data for each condition affects the accuracy of fault diagnosis.

Some researchers have used data processing techniques to preprocess the collected data for multiple working conditions and then treat the processed data as a single working condition for fault diagnosis [17]–[19]. Gu *et al.* [17] used directional entropy weighted kernel entropy component analysis to perform fault diagnosis on reduced dimensional multiple working conditions data, and the method does not guarantee real-time fault diagnosis. Zhao *et al.* [18] used Batch Normalization (BN) in a Convolutional Neural Network (CNN) to eliminate the distribution differences of multiple working conditions data. Wei *et al.* [19] reduce the impact of multiple working conditions on fault diagnosis results by expanding the dimensionality of CNN convolution kernels.

The above researches reduce the influence of data quality problems of multiple working conditions on fault diagnosis results to a certain extent, but the fault diagnosis method without distinguishing working conditions does not make full use of data information of different working conditions, which affects the accuracy of fault diagnosis. Therefore, Jacobs *et al.* [20] introduced the idea of ''modularity'' into the design of artificial neural networks for the first time, which improved the generalization ability of neural networks. Modular neural networks include two stages, module division and module combination. In terms of module division, Bo *et al.* [21] used fuzzy c-means to divide the measurement space into multiple modules. Geng *et al.* [22] used wavelet transform to classify the extracted features to obtain different modules. Although the modular approach described above improves the generalization of the network, there is a problem of information loss during the combination of modules. Therefore, in terms of module combination. Sabour *et al.* [23] proposed a capsule network model based on dynamic routing rules, which makes each capsule express different features by forming multiple neurons into an output capsule and building the location relationship of different features through a dynamic routing algorithm, which makes the network have a stronger feature representation capability. Existing dynamic routing methods use the top layer features with the highest level of abstraction as the basis for fault diagnosis, without taking into account the loss of information during the layer-by-layer extraction of neural network features.

Features on different layers of the neural network represent multiscale features with different levels of abstraction, and many shallow scale features containing fault information are lost in the process of feature extraction, which results in underutilization of the data. Shao *et al.* [24] used Local Preserving Projection (LPP) method to fuse multiscale features of DNN to improve the accuracy of fault diagnosis compared with the traditional layer-by-layer feature extraction method. Deep Belief Networks (DBN) and DNN have similar structure, so Li *et al.* [25] used LPP to fuse multiscale features of DBN to improve the feature extraction capability of the network. However, the use of a separate fusion algorithm will make the network more complex and cause difficulties in network convergence. Zhu *et al.* [26] designed a multiscale CNN network structure by splicing the convolutional and pooling layer features in the CNN into a multiscale feature matrix, which is connected to the fully connected layer to keep the global and local information synchronized and

improve the feature extraction capability of the network. Zhou *et al.* [27] designed a feature fusion network to fuse the features on different layers of DNN for fault diagnosis, which reduces the information loss during feature extraction and improves the diagnosis accuracy. However, the above methods would make the useless information at the shallow scale of the neural network interfere with the subsequent fault diagnosis, thus affecting the accuracy of the fault diagnosis.

The effect of deep learning neural network fault diagnosis depends on the quantity and quality of data and the utilization of data. Traditional neural networks have the problem of information loss in the process of layer-by-layer feature extraction, which will make the diagnosis results suffer. In fact, features represented in shallow scale may also contain useful information related to fault feature when system is operated in multiple working conditions. In fact, features represented in shallow scale may also contain useful information related to fault feature when system is operated in multiple working conditions. Therefore, this paper aims to design a new multiscale feature fusion method to reduce information loss during feature extraction and to achieve joint optimization between different working conditions, and thus improve the accuracy of multiple working condition fault diagnosis.

The innovations of this paper are as follows:

1) A cross working condition multiscale recursive feature fusion method is proposed to solve the problem of multiple working condition fault diagnosis in which traditional deep learning methods require a large number of working condition labels.
2) The network is first divided into single working condition modules and multiple working conditions modules that do not distinguish between conditions, and then features at the same scale are fused between the modules, followed by a recursive strategy designed to reduce information loss between scales.
3) The problem of multiple working conditions destroys the consistency of data distribution and makes the collected data of low quality. The traditional Multiple working condition fault diagnosis method treats multiple working conditions data as single working conditions data as the input of the neural network, which cannot make full use of the fault information contained in the shallow scale of the neural network. The proposed method makes full use of a small amount of data with working condition labels and a large amount of data without working condition labels, reduces the information loss in the process of neural network feature extraction, and can realize the implementation of accurate multiple working conditions fault diagnosis.

II. DNN RELATED THEORY

Autoencoder (AE) is a kind of unsupervised network that extracts features to restructure the input by encoding and decoding, which can effectively abstract the hidden features of the input. Since unsupervised AEs alone are weak in data feature extraction, DNN can be constructed by stacking multiple AEs. This stacking approach can extract features from the original input data layer-by-layer, and utilizes supervised parameter fine-tuning to optimize network performance. The DNN structure is shown in Fig. 1

III. A CROSS WORKING CONDITION MULTISCALS RECURSIVE FEATURE FUSION METHOD FOR FAULT DIAGNOSIS OF ROLLING BEARING IN MULTIPLE WORKING CONDITIONS

To solve the problems of low quantity of single working condition data, low quality of multiple working conditions data and information loss in the process of neural network feature extraction in rolling bearing multiple working condition fault diagnosis. This section proposes a deep learning-based fault diagnosis model to recursively fuse the multiscale feature on cross working conditions, such that data without working condition label can also be referred to train a satisfying deep learning model for fault diagnosis of bearing operated in multiple working conditions.

A. ANALYSIS OF PROBLEMS WHEN SEPARATE MODELLING OF DIFFERENT WORKING CONDITIONS IN MULTIPLE WORKING CONDITION DATA

The single working condition refers to the situation where the load is stable and the motor speed is stable during the operation of the equipment. The multiple conditions are the cases where the load changes during the operation of the equipment, which results in the collected data containing multiple speeds.

Since the fault samples collected during the operation of the equipment contain multiple working conditions, and the number of data obtained for separated working condition after distinguishing the working conditions is small, the accuracy of the fault diagnosis model based on single working condition data cannot be guaranteed. Traditional fault diagnosis methods based on deep learning have been used by some researchers to model multiple working condition samples as a single working condition when the number of samples is limited, in order to ensure that the number of samples is sufficient. Some researchers have also trained multiple deep neural networks by dividing the multiple working condition data into multiple single working condition datasets.

In order to verify the problem of data quality for multiple conditions on the fault diagnosis results, this section used DNN to do pre-analysis experiments on single and multiple working condition data with fault size of 0.007 inches. Results of the experiments are compared as listed in Table 1.

It can be seen from Table 1 that the fault diagnosis accuracy of single working condition diagnosis is 13.75% higher than that of multiple working condition diagnosis due to the difference in distribution between the data of different working conditions when using the same fault diagnosis method, which would cause difficulties in feature extraction

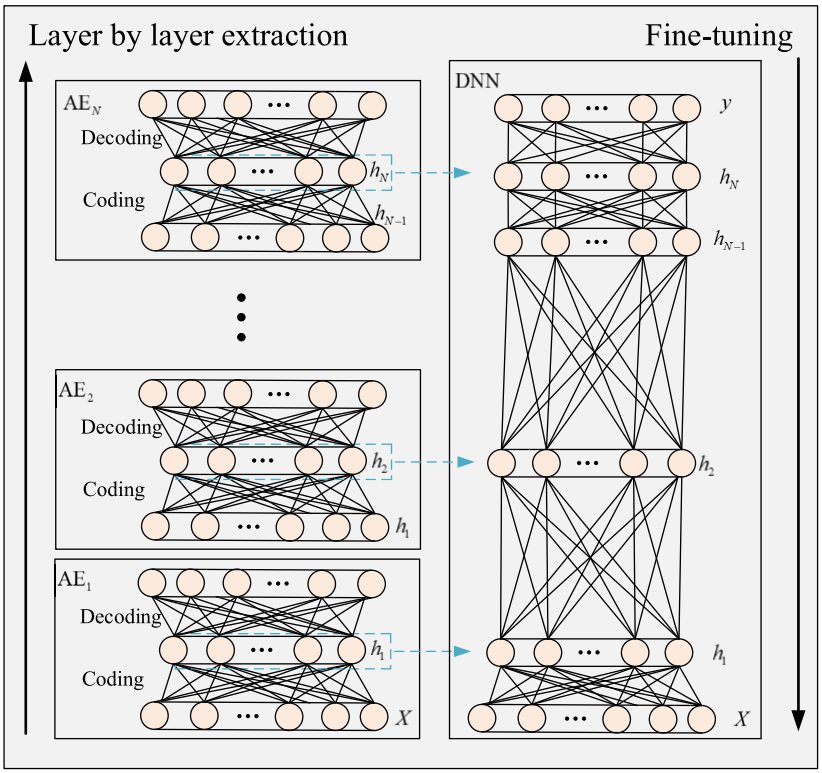

FIGURE 1. Structure diagram of DNN.

TABLE 1. Experiments with fault diagnosis using single working condition data and multiple working condition data when the fault diagnosis methods are the same.

Experiment	Single working condition	Multiple working condition
Motor Load (HP)	3	0/1/2/3
Working condition variability	---	The variability of working conditions is not considered.
Fault type	Inner race /Outer race / Ball /Normal condition	Inner race /Outer race / Ball /Normal condition
Number of training set samples	800	800
Number of test set samples	400	400
Training set accuracy	100%	100%
Test set accuracy	85.25%	71.50%

if the difference between such working conditions is not taken into account.

In order to improve the readability of the experimental results, this section gives the confusion matrix of fault diagnosis using DNN for single and multiple working condition data, as shown in Fig. 2. The labels in the confusion matrix are the fault types in row 4 of Table 1. It can be seen from Fig. 2 that the subfigure (b) has more misclassified samples for multiple working condition fault diagnosis. That is because direct fault diagnosis for multiple working conditions data without taking into account the diversity of operating conditions can affect the effectiveness of the fault diagnosis model.

In order to verify the different distinguishability of faults in the data for different working conditions. In this section, fault diagnosis comparison experiments were designed for four working conditions with fault sizes of 0.007 inches, the experimental design is listed in Table 2.

Comparing columns 3 and 4 in Table 2 shows that there is a 40.63% difference in fault diagnosis accuracy between working condition 2 and working condition 3, which is due to the different degrees of distinguishability of data faults for different working conditions. Comparison of column 4 with columns 2 and 5 of Table 2 also shows that there are significant differences in diagnostic accuracy due to different degree of fault distinguishability. Fig. 3 shows the different results of fault diagnosis for different working conditions.

B. A CROSS WORKING CONDITION MULTISCALE RECURSIVE FEATURE FUSION METHOD FOR FAULT DIAGNOSIS IN MULTIPLE WORKING CONDITIONS

The existing multiple working condition fault diagnosis method requires an advance division of the working conditions before fault diagnosis, which will lead to the problem of small number size of training samples for separated

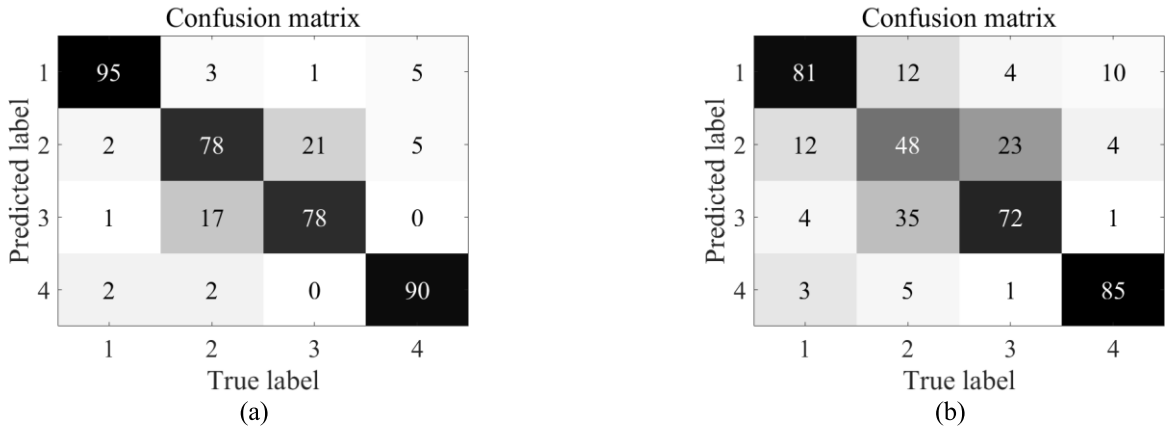

FIGURE 2. Confusion matrix for fault diagnosis using DNN for single working condition data and multiple working condition data (a) Confusion matrix for single working condition fault diagnosis and (b) Confusion matrix for multiple working condition fault diagnosis.

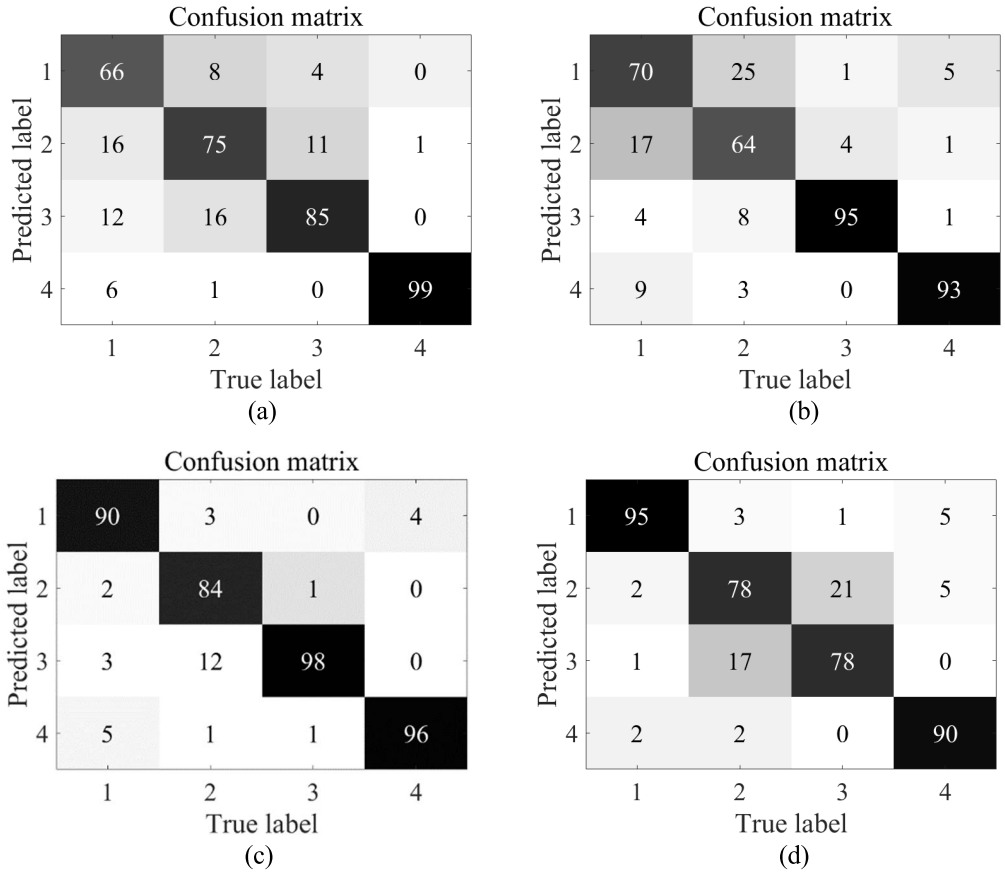

FIGURE 3. Confusion matrix for fault diagnosis of different working conditions (a) Working condition 1 (b) Working condition 2 (c) Working condition 3 and (d) Working condition 4.

single working condition fault diagnosis model. Therefore, this section proposes a deep learning based fault diagnosis model to recursively fuse the multiscale feature on cross working conditions to reduce information loss during feature extraction and to achieve joint optimization between different working conditions, and thus improve the accuracy of multiple working condition fault diagnosis.

1) DESIGN OF A MULTIPLE WORKING CONDITION FAULT DIAGNOSIS ALGORITHM WITH CROSS WORKING CONDITION MULTISCALE RECURSIVE FUSION

In order to fully extract the data features when the number of samples with working condition labels is limited, so as to obtain more accurate multiple working condition fault diagnosis results. This section first fused features at the

same scale from different working condition modules, then fused feature with the same scale from multiple working condition modules, and then feeds the fused features back to each module for feature extraction at the next scale. Recursive propagation in this modular comprehensive manner enables joint optimization between different modules. Finally, a dynamic routing approach is used for fault diagnosis of the recursively fused features to achieve accurate multiple working condition fault diagnosis. The steps of multiple working condition fault diagnosis based on multiscale recursive fusion cross working conditions are as follows.

Step1: Module division of multiple working conditions

The multiple working conditions data collected when the rolling bearing is running in Q working conditions is $X = \{X_1, X_2, \cdots, X_q, \cdots, X_Q, X_{\text{All}}\}$, where $X_1, X_2, \cdots, X_q, \cdots, X_Q$ are the fault data with working condition labels, X_q indicates the data of the q-th working conditions, and X_{All} is the fault data without working condition label. Different modules are established according to the working condition labels, and the data X_{All} without working condition labels are divided into multiple working condition modules as shown in (1).

$$X \rightarrow \{\text{DNN}_1, \text{DNN}_2, \cdots, \text{DNN}_q, \cdots, \text{DNN}_Q, \text{DNN}_{\text{All}}\} \tag{1}$$

where $\text{DNN}_1, \text{DNN}_2, \cdots, \text{DNN}_q, \cdots, \text{DNN}_Q$ denotes the network in the different working condition modules and DNN_{All} denotes the DNN network in the indistinguishable working condition modules.

Step2: Design of algorithms for fusion of single scale features cross working conditions

Firstly, the multiple working conditions data $X_1, X_2, \cdots, X_q, \cdots, X_Q$ are used as the input of different modules, and the first hidden layer features $h_{1,1}, h_{1,2}, \cdots, h_{1,q}, \cdots, h_{1,Q}$ of Q conditions are extracted using the DNN network in the module, and the different hidden layer features of the neural network are multiscale features with different degrees of abstraction. For the multiple working conditions data X_Q without condition labels, the first scale features $h_{1,All}$ are extracted using DNN_Q. The feature fusion of the first scale features of the different working conditions is shown in (2)-(5).

$$\text{Feature}_{\text{fu},1} = \sigma\left(W_{\text{fu},1}\text{Dim}_{L,1} + b_{\text{fu},1}\right) \tag{2}$$

$$\text{Dim}_{L,1} = G_{\text{fu}}\left(\text{Feature}_{g,1}, h_{1,\text{All}}\right) \tag{3}$$

$$\text{Feature}_{g,1} = \sigma\left(W_{g,1}\text{Dim}_{L,1,Q} + b_{g,1}\right) \tag{4}$$

$$\text{Dim}_{L,1,Q} = G_{\text{fu}}\left(h_{1,1}, h_{1,2}, \cdots, h_{1,Q}\right) \tag{5}$$

where $\text{Feature}_{\text{fu},1}$ denotes the first scale cross-condition fusion feature, σ denotes the nonlinear activation function, $W_{\text{fu},1}, b_{\text{fu},1}$ denotes the weight and bias, $\text{Dim}_{L,1}$ denotes the expanded dimensional features of the first scale of multiple working conditions data, and G_{fu} is the feature combination strategy, which is directly spliced in this algorithm to reduce the computational effort. $\text{Feature}_{g,1}$ is a feature of the first scale containing information on the different working conditions, $\text{Dim}_{L,1,Q}$ denotes the expanded dimensional features of the first scale of different working conditions.

Step3: Design of a recursive method cross working conditions between different scales.

In order to use the features with a higher level of fault distinguishability for optimal fault diagnosis in other working conditions and to reduce the problem of information loss in the neural network feature extraction process kind, a recursive method cross working conditions between different scales is designed. For the first and second scales, the fusion features $\text{Feature}_{\text{fu},1}$ are first fed back into the different working conditions module and the multiple working conditions module as shown in (6)-(7).

$$\begin{aligned}\text{Feature}_{\text{fb2},q} &= G_{\text{fu}}\left(\text{Feature}_{\text{fu},1}, h_{1,q}\right) \\ &= \left\{\begin{array}{l} G_{\text{fu}}\left(\text{Feature}_{\text{fu},1}, h_{1,1}\right), \cdots, \\ G_{\text{fu}}\left(\text{Feature}_{\text{fu},2}, h_{1,1}\right), \cdots, \\ G_{\text{fu}}\left(\text{Feature}_{\text{fu},q}, h_{1,1}\right), \cdots, \\ G_{\text{fu}}\left(\text{Feature}_{\text{fu},1}, h_{1,Q}\right) \end{array}\right\}\end{aligned} \tag{6}$$

$$\text{Feature}_{\text{fb2,All}} = G_{\text{fu}}\left(\text{Feature}_{\text{fu},1}, h_{1,All}\right) \tag{7}$$

where $q \in (0, Q]$. The feature $\text{Feature}_{\text{fb1,Q}}$, $\text{Feature}_{\text{fb1,All}}$ is used to optimize the feature extraction at the second scale of each module, as shown in (8)-(9).

$$h_{2,q} = \sigma\left(W_{2,q}\text{Feature}_{\text{fb},2} + b_{\text{fd},2}\right) \tag{8}$$

$$h_{2,All} = \sigma\left(W_{\text{fb2,All}}\text{Feature}_{\text{fb2,All}} + b_{\text{fb2,All}}\right) \tag{9}$$

where $h_{2,q} = \left[h_{2,1}, h_{2,2}, \cdots, h_{2,Q}\right]$, $W_{2,q}, W_{2,\text{All}}$ and $b_{2,q}, b_{2,\text{All}}$ are the coding weight parameters and bias parameters for the second scale feature extraction, respectively. As the second scale feature extraction process for each network adds the fusion feature $\text{Feature}_{\text{fu},1}$ feedback, the second scale features extracted are better than those extracted using only single working condition data.

The second scale features of the different modules are then fused cross the working conditions as shown in (10).

$$\text{Feature}_{\text{fu},2} = \text{SinglescaleFu}\left(h_{2,q}, h_{2,\text{All}}\right) \tag{10}$$

where $\text{SinglescaleFu}[\cdot]$ denotes the single scale cross working condition feature fusion algorithm designed in Step2. $h_{2,1}, h_{2,2}, \cdots, h_{2,q}, \cdots, h_{2,Q}$ denotes the second-scale features of the different modules and $\text{Feature}_{\text{fu},2}$ denotes the fusion features of the second-scale features. The $\text{Feature}_{\text{fu},2}$ is then fed back to each module for third-scale feature extraction and the third-scale features from each module are fused cross the working conditions. As shown in (11).

$$\begin{aligned}&\text{Feature}_{\text{fu},3} \\ &= \text{SinglescaleFu}\left(g\left(G_{\text{fu}}\left(\text{Feature}_{\text{fu},2}, h_{2,q}, h_{2,\text{All}}\right)\right)\right)\end{aligned} \tag{11}$$

where $g(\cdot)$ denotes the feature extraction function using AE. The multiscale features of different modules are recursively

TABLE 2. Experiments with fault diagnosis using different working conditions data when the fault diagnosis methods are the same.

Experiment	Working condition 1	Working condition 2	Working condition 3	Working condition 4
Motor Load (HP)	0	1	2	3
Fault type	Inner race /Outer race / Ball /Normal condition	Inner race /Outer race / Ball /Normal condition	Inner race /Outer race / Ball /Normal condition	Inner race /Outer race / Ball /Normal condition
Number of training set samples	800	800	800	800
Number of test set samples	400	400	400	400
Test set accuracy	81.25%	80.50%	92.00%	85.25%

fused n times in the same feature fusion method to obtain the n-th scale fusion feature $\text{Feature}_{\text{fu},n}$ as shown in (12).

$$\begin{aligned}&\text{Feature}_{\text{fu},n}\\&= \text{SinglescaleFu}\left(g\left(G_{\text{fu}}\left(\text{Feature}_{\text{fu},n-1}, h_{n-1,q}, h_{n-1,\text{All}}\right)\right)\right)\end{aligned} \tag{12}$$

where $\text{Feature}_{\text{fu},n-1}$ denotes the features after the $(n-1)$-th recursive fusion, $h_{n-1,q} = \left[h_{n-1,1}, h_{n-1,2}, \cdots, h_{n-1,q}\right]$ denotes the features at the $(n-1)$-th scale extracted by the different working condition modules. $h_{n-1,\text{All}}$ denotes the features at the $(n-1)$-th scale extracted by the multiple working condition modules that do not distinguish between working conditions.

The fusion features $\text{Feature}_{\text{fu},1}, \text{Feature}_{\text{fu},2}, \cdots, \text{Feature}_{\text{fu},n}$ at different scales are then combined into a $Feature_{FU}$ input classifier. The structure of the across working conditions multiscale recursive fusion network proposed in this paper is shown in Fig. 4.

Step4: Design of fault diagnosis classifier based on fusion features.

The feature extraction process is optimized by feature fusion for each modular network for multiple working conditions data. The combined features $\text{Feature}_{\text{FU}}$ of the fusion features $\text{Feature}_{\text{fu},1}, \text{Feature}_{\text{fu},2}, \cdots, \text{Feature}_{\text{fu},n}$ at different scales are then used as input to the fault diagnosis classifier. The feature combination process is shown in (13).

$$\text{Feature}_{\text{FU}} = \text{G}_{\text{fu}}\left(\text{Feature}_{\text{fu},1}, \text{Feature}_{\text{fu},2}, \cdots \text{Feature}_{\text{fu},n}\right) \tag{13}$$

where G_{fu} is the combined strategy of fusion features at different scales, and the multiscale fusion features $\text{Feature}_{\text{FU}}$ are batch normalized as shown in (14).

$$\text{Feature}_{\text{norm}} = \gamma \frac{\text{Feature}_{\text{FU}} - E[\text{Feature}_{\text{FU}}]}{\sqrt{\text{Var}[\text{Feature}_{\text{FU}}] + \varepsilon}} + \beta \tag{14}$$

where $E(\cdot)$ and $\text{Var}(\cdot)$ are the mean and standard deviation of the input $\text{Feature}_{\text{FU}}$, γ and β are trainable, and ε is a very small number to ensure that the denominator is not 0.

Dividing $\text{Feature}_{\text{norm}}$ into different vector neurons and adding the Squashing function after each vector neuron scales the length of the vector neuron to between 0 and 1 and does not change the direction of the vector. The various vector neurons are combined through a dynamic routing strategy, as in (15)-(17).

$$v_m = \sum\nolimits_m c_{\text{nm}} \hat{t_{m|n}} \tag{15}$$

$$\hat{t_{m|n}} = W_{\text{nm}} t_n \tag{16}$$

$$c_{\text{nm}} = \frac{\exp(t_{\text{nm}})}{\sum_m \exp(t_{\text{nm}})} \tag{17}$$

where m is the number of classes of faults, W_{nm} is the weight matrix, and the product of the two is used to calculate the prediction $\hat{t_{m|n}}$, $\sum_m c_{\text{nm}} = 1$, and c_{nm} is updated by a dynamic routing algorithm with $t_{\text{nm}} = 0$ at the initial iteration.

The predicted output v_c is obtained by adding the Squashing function after the fused v_m, as shown in (18).

$$v_c = f_s(v_m) = \frac{||v_m||^2}{1 + ||v_m||^2} \frac{v_m}{||v_m||} \tag{18}$$

where v_c denotes the predicted fault type, and the global network is optimized by setting the minimization loss function in the network training to achieve fault diagnosis. The structure diagram of the fault diagnosis classifier is shown in Fig. 5.

Step5: Design global optimization strategy to fine-tune each network parameters

The relatively small amount of data for single working conditions after dividing the working conditions leads to difficulties in feature extraction using neural network. By using a multiscale recursive fusion method to extract features cross working conditions and using a global optimization strategy to fine-tune parameters in reverse, joint optimization of samples with and without working conditions labels can be achieved. The global optimization strategy designed in this paper is shown in Fig. 6.

The loss function is constructed via the predicted value v_c of the recursive fusion network, as shown in (19).

$$\begin{aligned}J_{\text{dr}} = {} & T_c \max(0, m^+ - ||v_c||)^2 \\ & + \lambda(1 - T_c) \max(0, ||v_c|| - m^-)^2\end{aligned} \tag{19}$$

where T is the target class of the current fault. $m^+ = 0.9$ and $m^- = 0.1$. λ is the loss weight to reduce the failure to occur, and in this paper, λ is set to 0.6. In order to adjust the parameters of the multi working conditions module

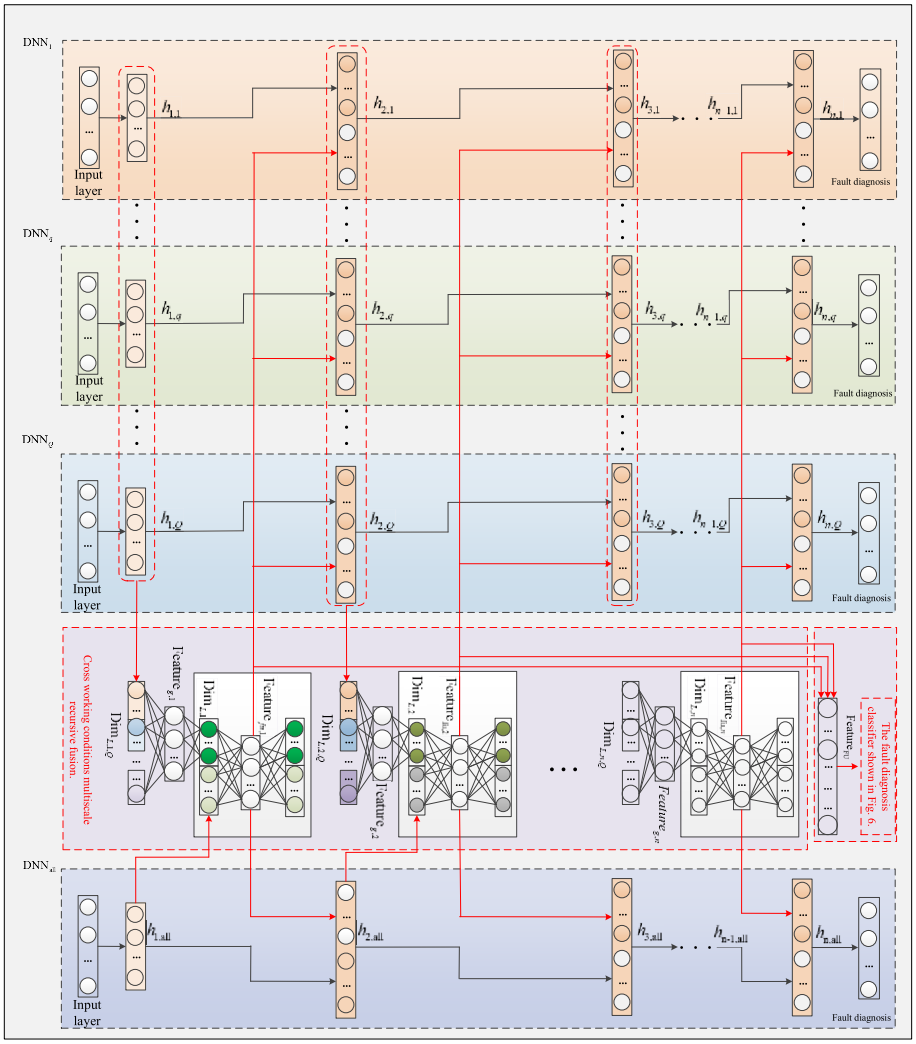

FIGURE 4. Structure diagram of cross working conditions multiscale recursive fusion neural network.

network simultaneously, so the Softmax layer is added to the loss function to update the parameters of the multi working conditions module network simultaneously while ensuring the update of the global feature extraction for the purpose of global optimization, and the loss function J_S is constructed by comparing the output label of Softmax with the real label as shown in (20).

$$J_S = -\frac{1}{K}\sum \text{label}_{\text{real}} \ln(\text{label}_{\text{output}}) + (1 - \text{label}_{\text{real}}) \ln(1\text{-label}_{\text{output}}) \quad (20)$$

The local network of each module can be optimized simultaneously by minimizing the global error loss function, and the global loss function is shown in (21).

$$J(\theta) = J_{\text{dr}} + J_S \quad (21)$$

The global error E_{global} is calculated from the global error loss function $J(\theta)$ and when E_{global} is fed back into the fusion network, the error feedback is shown in (22).

$$\begin{aligned} E_{\text{global}} &= E_{\text{fu}} + E_{\text{model},1} + E_{\text{model},2} + \cdots \\ &\quad + E_{\text{model},q} + \cdots + E_{\text{model},Q} + E_{\text{model,All}} \\ &= E_{\text{fu}} + \left(E_{\text{DNN},1} + E_{fu,1}\right) + \left(E_{\text{DNN},2} + E_{\text{fu},2}\right) + \cdots \\ &\quad + \left(E_{\text{DNN},q} + E_{\text{fu},q}\right) \\ &\quad + \left(E_{\text{DNN},Q} + E_{\text{fu},Q}\right) + \left(E_{\text{DNN,All}} + E_{\text{fu,All}}\right) \end{aligned} \quad (22)$$

where E_{fu} denotes the error of the fusion network. $E_{\text{model},q}$ denotes the error of the q-th module. $E_{\text{model},q}$ consists of the error $E_{\text{DNN},q}$ generated by the DNN network for that module and the error $E_{\text{fu},q}$ returned by the fusion network to the q-th network. And for the DNN network in the q-th working condition module, the error $E_{\text{DNN},q}$ is obtained by minimizing the loss function, which is shown in (23).

$$J_{\text{DNN},q} = -\frac{1}{K}\sum \text{label}_{\text{real},q} \ln(\text{label}_{\text{output},q}) + (1 - \text{label}_{\text{real},q}) \ln(1 - \text{label}_{\text{output},q}) \quad (23)$$

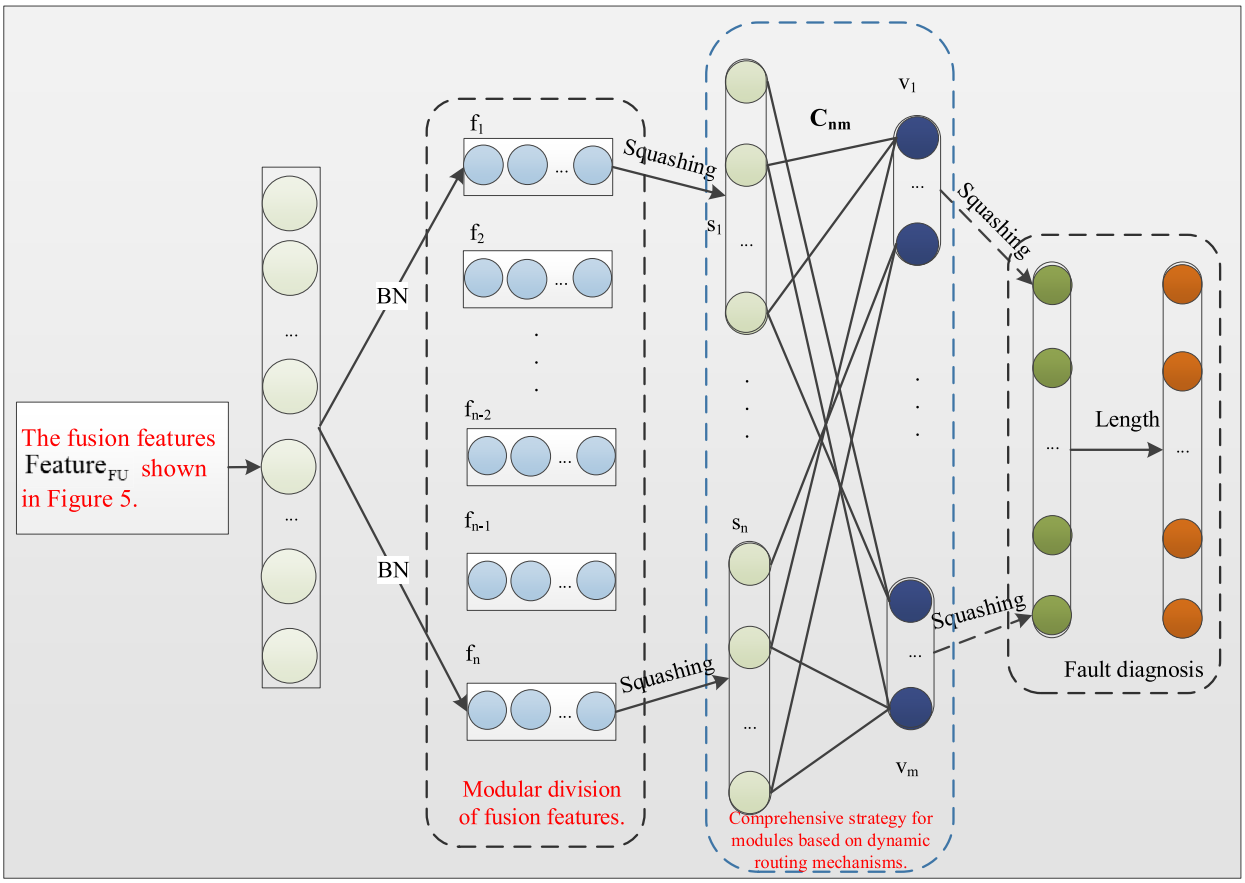

FIGURE 5. Structure diagram of the fault classifier based on fusion features.

Through global parameter fine-tuning, joint optimization between different modules can be achieved to obtain the trained network parameters as shown in (24).

$$\mathrm{Tr}_{\mathrm{global}} = \mathrm{Train}(\mathrm{Net}_{\mathrm{DNN},1}, \mathrm{Net}_{\mathrm{DNN},2}, \ldots, \mathrm{Net}_{\mathrm{DNN},Q}, \mathrm{Net}_{\mathrm{Fedration}}; J(\theta);\ X) \quad (24)$$

where $\mathrm{Tr}_{\mathrm{global}} = \{\theta_{\mathrm{DNN},1}; \theta_{\mathrm{DNN},2}; \ldots; \theta_{\mathrm{DNN},Q}; \theta_{\mathrm{DNN,All}}; \theta_{\mathrm{fed}}; \theta_{\mathrm{Softmax}}; \theta_{\mathrm{dr}}\}$ is the trained parameters for each network.

2) MULTIPLE WORKING CONDITION FAULT DIAGNOSIS BASED ON CROSS WORKING CONDITION MULTISCALE RECURSIVE FUSION NEURAL NETWORKS.

When online data $X(k)$ are collected at time k, the data $X(k)$ without working condition labels are fed into a trained cross working conditions multiscale recursive fusion neural network with global fine-tuning (G-CMRFNN) for feature extraction, as shown in (25).

$$\mathrm{Feature}_{\mathrm{FU}}(k) = G_{\mathrm{fu}}\left(\mathrm{Net}_{\mathrm{G\text{-}CMRFN}}, \mathrm{Tr}_{\mathrm{global}}, \mathrm{X}_{\mathrm{online}}(k)\right) \quad (25)$$

The feature $\mathrm{Feature}_{\mathrm{FU}}(k)$ is fed into the classifier for fault diagnosis as shown in (26) and (27).

$$h_{\theta,\mathrm{G\text{-}CMRFN}}(k) = \begin{bmatrix} p(\mathrm{label}=1)\ |\mathrm{Feature}_{\mathrm{FU}}(k); \theta_{\mathrm{fusion}}) \\ p(\mathrm{label}=2)|\mathrm{Feature}_{\mathrm{FU}}(k); \theta_{\mathrm{fusion}}) \\ \vdots \\ p(\mathrm{label}=L)|\mathrm{Feature}_{\mathrm{FU}}(k); \theta_{\mathrm{fusion}}) \end{bmatrix} = \frac{1}{\sum_{l=1}^{L} \theta_{fl}^{T}\mathrm{Feature}_{\mathrm{FU}}(k)} \begin{bmatrix} e^{\theta_{f1}^{T}\mathrm{Feature}_{\mathrm{FU}}(k)} \\ e^{\theta_{f2}^{T}\mathrm{Feature}_{\mathrm{FU}}(k)} \\ \vdots \\ e^{\theta_{\mathrm{fL}}^{T}\mathrm{Feature}_{\mathrm{FU}}(k)} \end{bmatrix} \quad (26)$$

$$\mathrm{lable}(k) = \underset{\mathrm{k=1,2,\,L,\,K}}{\mathrm{argmax}} \{h_{\theta_{\mathrm{fusion}},\mathrm{G\text{-}CMRFN}}(k)|X_{\mathrm{online}}(k); \theta_{\mathrm{fusion}})\} \quad (27)$$

where θ_{fusion} is the parameter of the multiscale recursive fusion network and $\mathrm{lable}(k)$ is the online fault diagnosis result at time k. The flowchart of G-CMRFNN algorithm is shown in Fig. 7.

IV. EXPERIMENT AND ANALYSIS

In the actual fault diagnosis application, the historical data collected during the operation of the equipment are

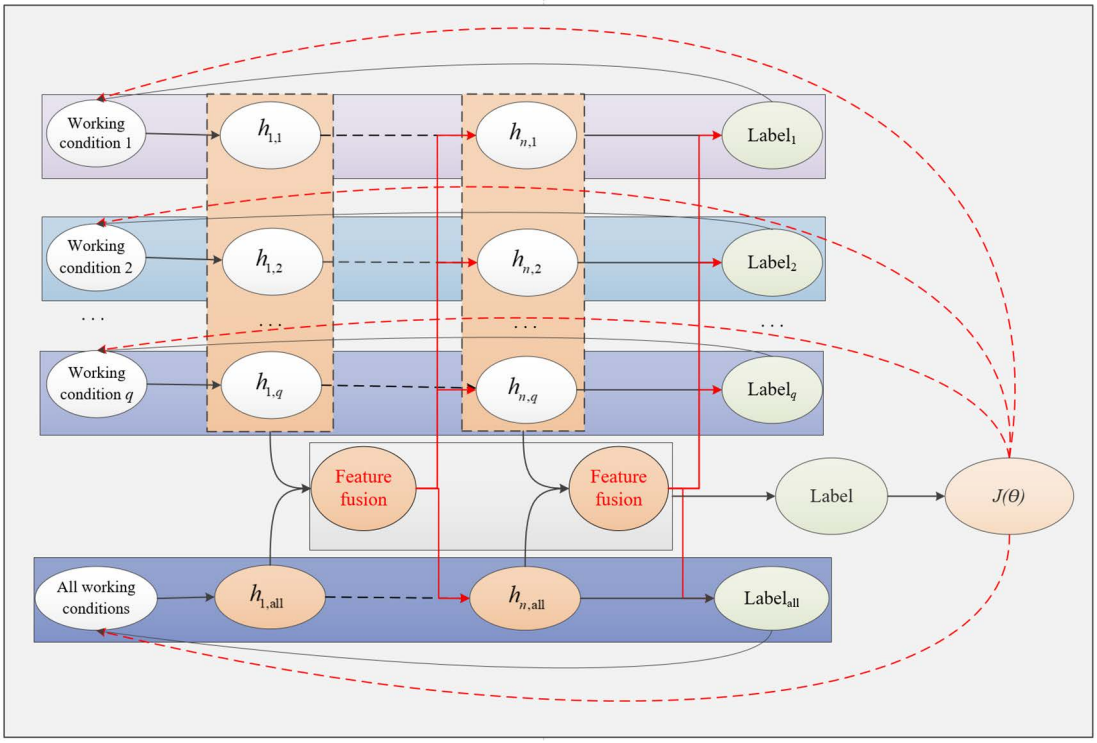

FIGURE 6. Global optimization strategy.

used to train the model, and then real-time fault diagnosis is performed on the online data. In order to verify the effectiveness of the algorithm, samples from different time periods of the experimental platform are considered as historical data and online data in the simulation experiments. This section verifies the effectiveness of the algorithm by using CWRU rolling bearing data [28].

A. EXPERIMENTAL DATA DESCRIPTION

The CWRU experimental platform is shown in Fig. 8. A single point of failure is introduced to the bearing by Electrical Discharge Machining (EDM). The vibration data is measured by an accelerometer placed on the drive side of the motor. CWRU data provides a wealth of available experimental data, for the outer race failure data at the 6 o'clock position, CWRU provides failure data for three sizes (0.007, 0.014, and 0.021 inch) at different loads. For the outer race faults at the 3 o'clock and 12 o'clock positions, CWRU provides fault data for two sizes (0.007 and 0.021 inch) under different loads. To verify the performance of the proposed method at different fault sizes, this section chose to use the fault data for the outer race at the 6 o'clock position. The sampling frequency of the data is 12 KHz, and the sampling time of the data is about 10s. In this section, the signals of the first 5s are considered as historical data. the signals of 5s-10 s are considered as online data.

The basis of the neural network training decision is the requirement that the training samples are independently and identically distributed. This section uses the common sliding window method to obtain samples with a window size of 900 and a sliding step size of 20. If the window is too small, the similarity between different samples is too strong and it will not have the meaning of independent samples, and if the window is too long, the similarity between samples will be weakened and the meaning of identical distribution will be lost.

There are 4 fault types with 3 fault size, as listed in Table 3. It can be seen that there are total of 10 fault types to be diagnosed. The load for different working conditions is 0-3HP, respectively. When the fault occurred in rolling bearings, the underlying statistical characteristics such as mean and variance of the vibration signals are different. Although these differences can support anomaly detection, they are not sufficient to distinguish different types of faults, so neural networks are needed to transform the vibration signals into a higher dimensional and more complex nonlinear space, thus making the different types of faults more distinguishable.

B. EXPERIMENTAL DETAIL

A neural network usually contains training set, validation set and test set, where the training set is used to train the neural network. The validation set is used to verify the model effect during the training process and helps to select the optimal model parameters. The test set is used to evaluate the accuracy of the model. However, the validation set is not mandatory,

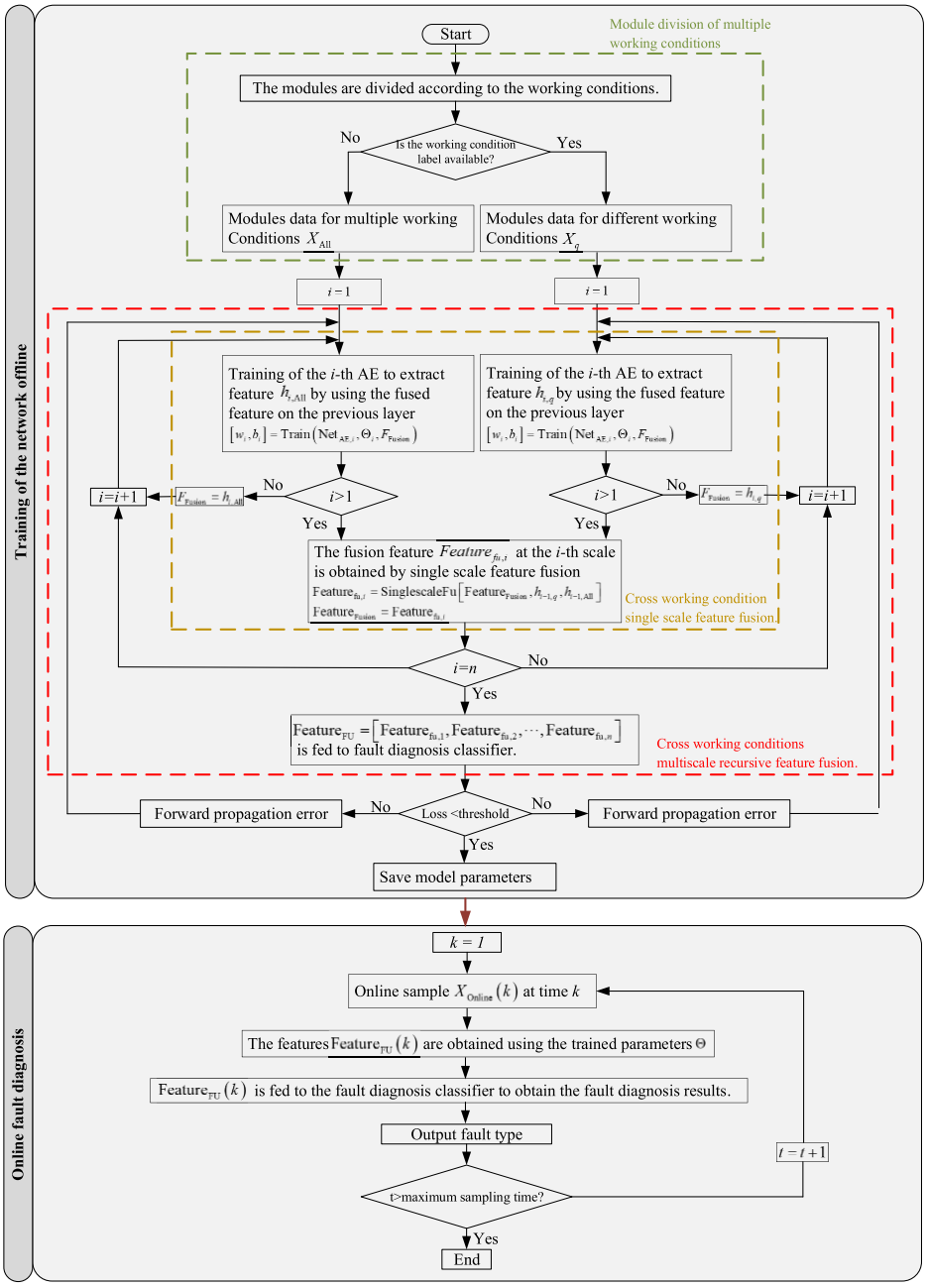

FIGURE 7. Flowchart of cross working conditions multiscale recursive fusion neural network.

this section use the training set to train the neural network model, end the training when the loss of the model reaches a threshold, and then use the test set to evaluate the accuracy of the model.

In order to verify the effectiveness of the algorithm the fault diagnosis experiments with multiple working conditions are designed as listed in Table 4. In Experiments 1-3, for different fault sizes, the ratio of the number of samples with and without working condition labels in the training set is 1:40. In order to verify the influence of the number of samples with working condition labels on the diagnosis results, Experiments 4-6 are designed, keeping the number of samples without working condition labels unchanged, and setting the ratio of the number of samples with and without working condition labels to 1:50. In order to verify the influence of the number of samples with working condition labels on the diagnosis results, Experiment 7 is designed. Experiments 7-9 were designed to verify the effect of the sample size without the working label on the diagnosis results, keeping the sample size with the working label unchanged and setting the ratios as 1:40, 1:50 and 1:60 respectively. In order to verify the effectiveness of the

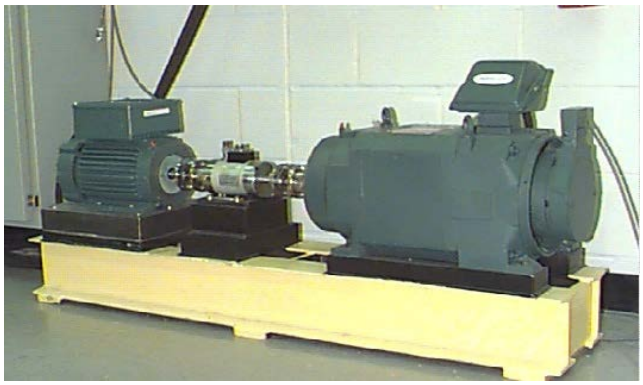

FIGURE 8. Experimental platform for the rolling bearing to obtain vibration signal [28].

TABLE 3. Detail of the experiment data.

Fault types	Fault size (Inches)	Label
Ball	0.007	1
Inner race	0.007	2
Outer race	0.007	3
Ball	0.014	4
Inner race	0.014	5
Outer race	0.014	6
Ball	0.021	7
Inner race	0.021	8
Outer race	0.021	9
Normal	0	10

algorithm when there are more working conditions, the data at the fan end of different loads are designed as different working conditions for experiments 10-12. For the test set, 400 samples were set for each type of fault.

Most of the current neural network models normalize the data before the sample data is input to the neural network, and normalization can make the neural network converge more easily. This section uses the maximum-minimum normalization method to normalize the data, as shown in (28).

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{28}$$

where x denotes the sample data and x^* denotes the normalized sample data. $x_{\max}$ and $x_{\min}$ denote the maximum and minimum values of the signal in the sample.

The experiment design is listed in Table 4. The relevant experimental model is listed in Table 5. The network hyperparameters for each network model are listed in Table 6.

C. ANALYSIS OF EXPERIMENTAL RESULTS

In this paper, the running environment of the experiment is Python 3.7.11. Tensorflow 2.3.0 GPU. Computer configuration is 11th Gen Intel(R) Core(TM) i9-11900K 3.50 GHz, GPU is NVIDIA GeForce RTX3090. The operating system is Windows 10 Home. The experimental results are listed in Table 7.

Comparing column 2 and column 3 of row 8 in Table 7. Since MDNN fused features from the last two scales of traditional DNN, which reduced the problem of information loss during feature extraction to a certain extent, therefore the accuracy of multiple working condition fault diagnosis was 6.08% higher than that of traditional DNN. However, only the last two scale features are fused by MDNN. Actually, different conditions features may be represented on different layers of the deep neural network, so only the last two scales are used and the information is still not fully utilized.

From column 4 of Table 7, it can be seen that the multiple working condition fault diagnosis accuracy of MCNN is higher than that of DNN and MDNN. this is because MCNN utilizes multiscale information with different degrees of abstraction by splicing and fusing the convolutional and pooling layers. therefore, in row 8 of Table 7, the fault diagnosis accuracy of MCNN is 9.95% higher than that of DNN and 4.27% higher than that of MDNN. MCNN only uses to the last layer of convolution and pooling features, and if all the shallow-scale features are spliced and fused, the useless information in the shallow scale will affect the subsequent fault diagnosis results. In contrast, using recursive fusion to utilize multiscale features can avoid the loss of useful information and the propagation of useless information. Therefore, it can be seen from column 5 of Table 7 that the fault diagnosis accuracy of MRFDNN is higher than that of MCNN.

It can be seen by comparing column 3 and column 5 of row 8 in Table 7. Due to the utilization of the multiscale recursive feature fusion method, MRFNN can fuse multiscale features on different layers of the DNN, thus achieving 16.78% higher accuracy than MDNN for multiple working condition fault diagnosis. Although MRFNN reduces the information loss during traditional DNN feature extraction by recursive fusion, and makes full use of multiscale features with different levels of abstraction. However, it does not take into account the different degree of distinguishability of faults between different working conditions. It can be seen from columns 5 and 6 in row 8 of Table 7 that, due to the use of the cross-condition multiscale recursive fusion method, CMRFNN can obtain more comprehensive fault features as it is fused with multiscale features from different condition modules, which makes CMRFNN 1.7% more accurate than MRFNN for multiple working conditions fault diagnosis. However, CMRFNN does not use a global optimization strategy to fine-tune the parameters of all module networks and therefore cannot achieve joint optimization between different working conditions. Comparing columns 6 and 7 in row 8 of Table 7, it is shown that the multiple working condition fault diagnosis accuracy achieves 94.85%. This is due to the global optimization strategy used by G-CMRFNN, which can realize the joint optimization of different working conditions and multiple working conditions.

Comparing column 2 and column 7 of row 8 in Table 7, it is shown that due to use of the cross-condition multiscale recursive fusion method, fusion of multiscale features on different working condition modules and multiple working condition modules, reduced information loss in the process

TABLE 4. Experiment design Related experimental models.

Experiment	Working Conditions	Number of samples of modules with different working conditions	Number of samples of multiple working conditions module without working condition division	Number of test set samples	Fault size (Inches)	Fault types
Experiment 1	Load(HP): 0/1/2/3 Sensor position: Drive end	200/200/ 200/200	8000	1600	0.007	Inner race /Outer race /Ball /Normal condition
Experiment 2	Load(HP): 0/1/2/3 Sensor position: Drive end	200/200/ 200/200	8000	1600	0.014	Inner race /Outer race / Ball /Normal condition
Experiment 3	Load(HP): 0/1/2/3 Sensor position: Drive end	200/200/ 200/200	8000	1600	0.021	Inner race /Outer race / Ball /Normal condition
Experiment 4	Load(HP): 0/1/2/3 Sensor position: Drive end	160/160/ 160/160	8000	1600	0.007	Inner race /Outer race / Ball /Normal condition
Experiment 5	Load(HP): 0/1/2/3 Sensor position: Drive end	160/160/ 160/160	8000	1600	0.014	Inner race /Outer race /Ball /Normal condition
Experiment 6	Load(HP): 0/1/2/3 Sensor position: Drive end	160/160/ 160/160	8000	1600	0.021	Inner race /Outer race / Ball /Normal condition
Experiment7	Load(HP): 0/1/2/3 Sensor position: Drive end	160/160/ 160/160	6400	4000	0.007/0.014/0.021	Inner race /Outer race / Ball /Normal condition
Experiment 8	Load(HP): 0/1/2/3 Sensor position: Drive end	160/160/ 160/160	8000	4000	0.007/0.014/0.021	Inner race /Outer race / Ball /Normal condition
Experiment 9	Load(HP): 0/1/2/3 Sensor position: Drive end	160/160/ 160/160	9600	4000	0.007/0.014/0.021	Inner race /Outer race / Ball /Normal condition
Experiment 10	Load(HP): 0/1/2/3 Sensor position: Drive end /Fan end	160/160/ 160/160/ 160/160/ 160/160	6400	4000	0.007/0.014/0.021	Inner race /Outer race / Ball /Normal condition
Experiment 11	Load(HP): 0/1/2/3 Sensor position: Drive end /Fan end	160/160/ 160/160/ 160/160/ 160/160	8000	4000	0.007/0.014/0.021	Inner race /Outer race / Ball /Normal condition
Experiment 12	Load(HP): 0/1/2/3 Sensor position: Drive end /Fan end	160/160/ 160/160/ 160/160/ 160/160	9600	4000	0.007/0.014/0.021	Inner race /Outer race / Ball /Normal condition

TABLE 5. Related experimental models.

Model abbreviation	Model explanation
DNN	Traditional deep neural network.
MDNN[27]	The last two scales of the traditional DNN are fused by splicing the features.
MCNN[26]	The last two scales of the traditional CNN are fused by splicing the features.
MRFNN	Multiscale recursive fusion neural network.
CMRFNN	Cross working conditions multiscale recursive fusion neural network.
G-CMRFNN	Cross working conditions multiscale recurrent fusion neural networks with global optimization.

of DNN feature extraction, and joint optimization between different modules through the global optimization strategy, G-CMRFNN has a higher accuracy of multiple working condition fault diagnosis than DNN by 21.17%, which proves the effectiveness of the method proposed in this paper.

Comparing rows 2 and 4 in Table 7 shows that for multiple working condition fault diagnosis with a fault size of 0. 007 inches, the number of samples with condition labels decreases, the lower the fault diagnosis accuracy, which is due to the fact that as the number of samples with condition labels decreases, fewer features of individual conditions are used for joint optimization. This indicates the effectiveness of joint optimization cross conditions. The same conclusion can be obtained from rows 3 and 6 and rows 4 and 7 in Table 7. Comparing rows 2-4 with rows 5-7 of Table 7 shows that in the four types of fault diagnosis with the same fault type and

TABLE 6. Network structure design.

Network	Number of layers	Learning rate	Activate function	Classifier	Optimization guidelines	Whether to perform global optimization
DNN	5 layers	0.001	Relu	Softmax	Error back propagation	Yes
MDNN	5 layers	0.001	Relu	Softmax	Error back propagation	Yes
MCNN	6 layers	0.001	Relu	Softmax	Error back propagation	Yes
MRFNN	6 layers	0.001	Squash/ Relu	Marginloss+Softmax	Dynamic routing strategy + error back propagation	Yes
CMRFNN	6 layers	0.001	Squash/ Relu	Marginloss+Softmax	Dynamic routing strategy + error back propagation	No
G-CMRFNN	6 layers	0.001	Squash/ Relu	Marginloss+Softmax	Dynamic routing strategy + error back propagation	Yes

TABLE 7. Fault diagnosis results when the number of samples with working condition labels is small.

Experiment	DNN	MDNN	MCNN	MRFNN	CMRFNN	G-CMRFNN
Experiment 1	81.33%	84.15%	88.45%	95.23%	95.65%	95.73%
Experiment 2	83.08%	88.40%	91.40%	95.90%	96.88%	97.05%
Experiment 3	86.70%	91.30%	93.13%	95.05%	96.13%	98.20%
Experiment 4	79.05%	81.30%	86.25%	93.03%	93.10%	94.18%
Experiment 5	80.88%	87.93%	88.30%	93.60%	95.18%	96.95%
Experiment 6	83.13%	89.75%	90.53%	94.73%	96.21%	97.78%
Experiment 7	73.68%	79.76%	83.35%	90.46%	92.16%	94.85%
Experiment 8	75.88%	83.33%	87.20%	91.00%	94.23%	95.30%
Experiment 9	77.78%	86.23%	89.33%	92.33%	95.70%	96.10%
Experiment10	64.40%	70.08%	74.35%	81.05%	84.03%	86.63%
Experiment 11	67.33%	72.20%	77.83%	83.18%	86.73%	88.58%
Experiment 12	69.73%	75.68%	80.20%	84.78%	87.40%	90.83%

different fault sizes, the fault distinguishability increases as the fault size increases and therefore the diagnosis accuracy of each fault diagnosis network improves.

Comparing row 8 and row 11 of Table 7, it can be seen that the more complex the working condition is, the lower the fault diagnosis accuracy is. This is because multiple working conditions destroy the distribution of data and lead to low data quality. The more complex the working conditions, the lower the data quality of the samples. And the effectiveness of deep learning depends on the quality of training samples, so the diagnostic accuracy of row 11 is lower than that of row 8. The same conclusion can be drawn by comparing row 8 with row 12 and row 9 with row 13. To improve the readability of the experimental results, the confusion matrix for the different network fault diagnosis results from Experiment 7 is given in Fig.9. The types of faults indicated by the labels in the figure are listed in Table3.

It can be seen from the subfigures in Fig.9 that for ball faults with a fault size of 0.014 inches in column 4, there are fewer samples correctly diagnosed for this type of fault compared to the other fault types in the multiple working condition fault diagnosis with different network models, which indicates a lower level of fault distinguishability.

Comparing subfigure (a) and subfigure (b) of Fig. 9, it can be seen that the 5th column of DNN diagnoses 35 more samples than the 5th column of MDNN, which is because the shallow scale contains useful information for fault diagnosis, and MDNN only uses the features of the latter two scales for splicing and fusion, which will have the problem of useful information loss, so although the overall accuracy of MDNN is higher than that of DNN, the diagnosis accuracy of individual faults may be reduced. The same problem exists in columns 1 and 2 of subfigure (b) and subfigure (c). The reason is that the information useful for fault diagnosis is lost during feature extraction. Using multiscale recursive approach can reduce the loss of useful information during feature extraction, and comparing subfigure d and column 5 of subfigure (a) shows that MRFNN diagnoses 49 more fault samples than DNN. Comparing column 5 of subfigure (e) and subfigure (f), it can be seen that MRFNN diagnoses 33 more fault samples than CMRFNN, which is due to the fact that CMRFNN does not fully utilize the information with and without working condition labeled data by global fine-tuning. Therefore, comparing column 5 of subfigure (g) and subfigure (e), it can be seen that G-CMRFNN can diagnose 8 more fault samples than MRFNN due to the use of global fine-tuning, which can make full use of the fault information in a small amount of working condition labeled data and a large amount of non-working condition labeled data, and can reduce the loss of useful information in the feature extraction process.

Comparing column 4 of subfigure (f) with subfigure (a), it can be seen that the G-CMRFNN proposed in this paper can correctly diagnose 341 fault samples when performing multiple working condition fault diagnosis due to the joint optimization using data without working condition labels and data with working condition labels, while the traditional DNN can only extract features of single working condition, so it

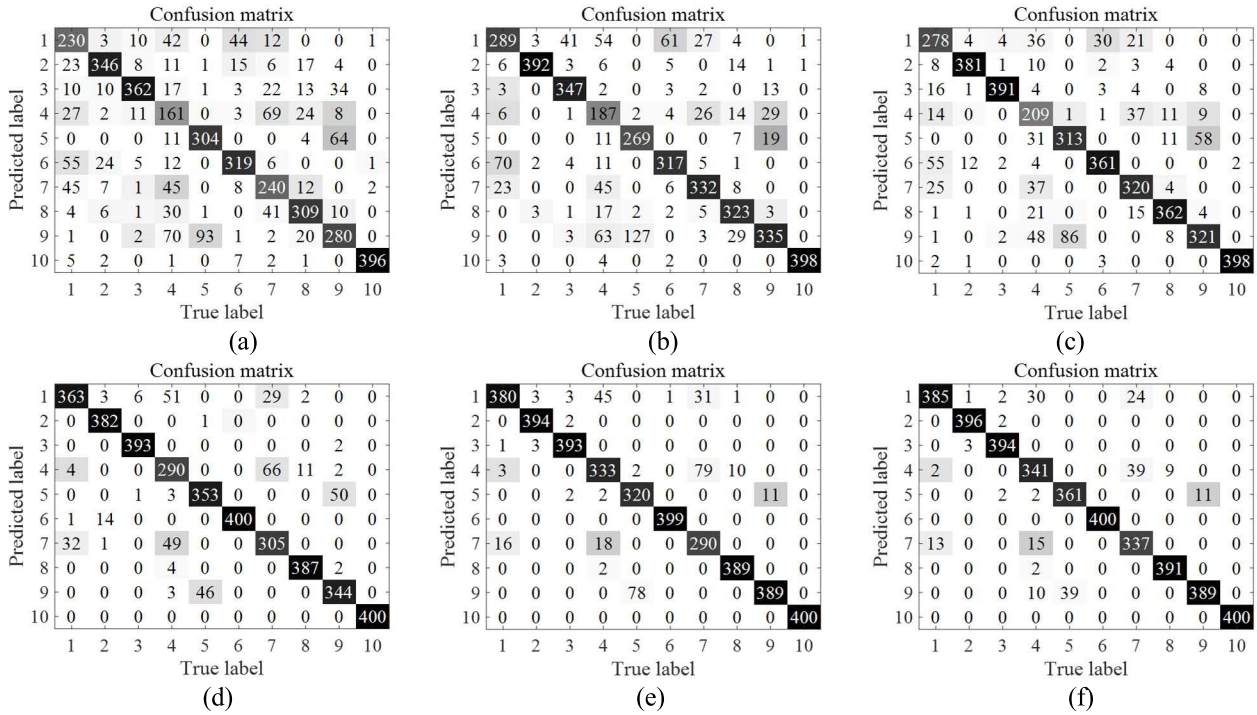

FIGURE 9. Fault diagnosis confusion matrix for different fault types with different fault sizes when the number of samples with working condition labels is 160 and the number of samples without working condition labels is 6000 (a) DNN (b) MDNN (c) MRFNN (d) CMRFNN and (e) G-CMRFNN.

TABLE 8. Offline training time and online diagnosis time for each model.

Time (s)	DNN	MDNN	MCNN	MRFNN	CMRFNN	G-CMRFNN
Offline training time	180.483	187.734	265.637	306.758	535.909	568.748
Online fault diagnosis time	0.015	0.016	0.170	0.210	0.349	0.352

is less effective in multiple working condition data, and the number of samples correctly diagnosed is 180 less than that of G-CMRFNN. Comparing column 4 of subfigure (f) and subfigure (b) shows that the MDNN only fused features from the last two scales and did not take into account the difficulty of feature extraction caused by multiple working conditions data. Therefore, it can diagnose 187 correct samples, 154 less than the G- CMRFNN. Comparing column 4 of subfigure (f) and subfigure (d) shows that although MRFNN reduces the information loss in the traditional DNN feature extraction process by recursive fusion, it makes full use of multiscale features with different degrees of abstraction. However, the multiscale features of other working conditions are not taken into account, which can affect the fault diagnosis results of multiple working conditions, and therefore 51 fewer fault samples are diagnosed than G-CMRFNN. Comparing column 4 of subfigure (f) and subfigure (e), it can be seen that CMRFNN utilizes the multiscale features in the different working condition modules due to the use of a multiscale recursive fusion scheme cross working conditions, but does not use global optimization to adjust the network parameters and cannot achieve joint optimization between different modules. However, G-CMRFNN achieves joint optimization cross different working conditions via global optimization, and therefore G-CMRFNN diagnoses 8 fault samples more than CMRFNN, which illustrates the effectiveness of G-CMRFNN.

To reflect the training complexity of the network, this section counts the average offline training time and online diagnosis time for each model, as listed in Table 8.

Comparing column 3, column 4 and column 5 in Table 8 shows that although both use multiscale feature fusion methods, recursive fusion utilizes information from all scales, so the offline training time and online diagnosis time are higher than the feature fusion methods that use only the last two scales. Comparing columns 6 and 7 shows that the diagnosis time is longer for the training time of G-CMRFNN than that of m through the adjustment of global optimization. Combined with the fault diagnosis accuracy in Table 7, when faced with fewer fault failure types, the existing methods can achieve more than 90% diagnosis accuracy, and perhaps there is no need to use the more advanced G. In the case of more fault types and complex working conditions, this paper argues that it is worthwhile to use the more advanced

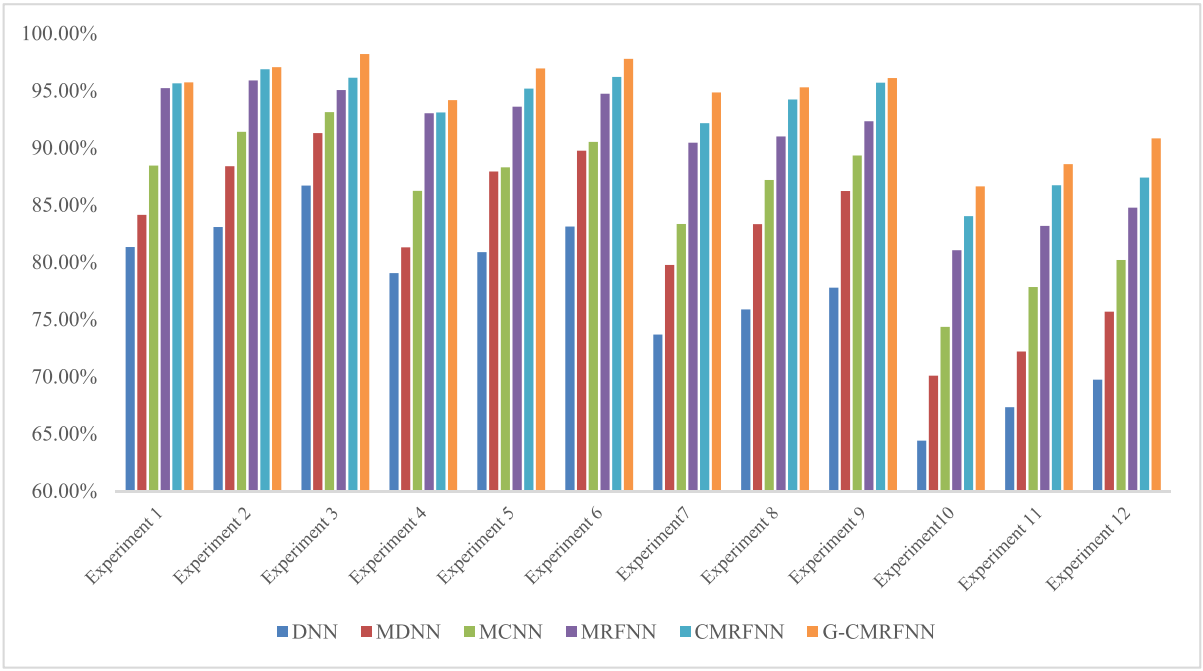

FIGURE 10. Histogram of experimental results.

G-CMRFNN and increase the computational complexity to obtain higher fault diagnosis accuracy. Fig.10 is a histogram of all experimental results.

V. CONCLUSION AND FUTURE WORK

Rolling bearings are one of the key components of electromechanical equipment, but the collected bearing data contains multiple working conditions. The multiple working conditions data destroys the assumption of i.i.d of data desired for the effectiveness of deep learning, which makes data feature extraction difficult. The traditional fault diagnosis method for multiple working conditions requires a large number of labeled samples, but existence of multiple working conditions makes it difficult to label the data. Therefore, this paper proposes a deep learning based fault diagnosis model to recursively fuse the multiscale feature on cross working conditions, such that data without working condition label can also be referred to train a satisfying deep learning model for fault diagnosis of bearing operated in multiple working conditions. Through designed cross working conditions multiscale recursive fusion strategy between different working condition modules, which can achieve the joint optimization of a large number of samples without working condition labels and a small number of samples with working condition labels, and achieve the full utilization of information. This method can improve the accuracy of multiple working condition fault diagnosis in the low quantity of single working condition data and the low quality of multiple working condition data, more fully utilize the limited quantity of labeled data, and achieve real-time accurate multiple working condition fault diagnosis.

The experimental results show that the proposed method outperforms the existing multiscale feature fusion method by more than 10% when the ratio of the number of data with and without condition labels is 1:40 for a large number of fault types. It is exciting that the diagnosis accuracy can be improved to more than 96% by increasing the number of samples without working labels when the number of samples with working labels is small and the fault types are complex.

This research has limitations when facing new working conditions that have not appeared in historical data, which may be contained in historical data from other companies, but data from different companies often cannot be shared directly. Therefore, multiple working condition fault diagnosis based on federated learning is a potential direction for future work.

REFERENCES

[1] S. M. K. Zaman and X. Liang, "An effective induction motor fault diagnosis approach using graph-based semi-supervised learning," *IEEE Access*, vol. 9, pp. 7471–7482, 2021.
[2] W. Yu and P. Lv, "An end-to-end intelligent fault diagnosis application for rolling bearing based on mobilenet," *IEEE Access*, vol. 9, pp. 41925–41933, 2021.
[3] T. A. Shifat and J.-W. Hur, "ANN assisted multi sensor information fusion for BLDC motor fault diagnosis," *IEEE Access*, vol. 9, pp. 9429–9441, 2021.
[4] D. Wang, K.-L. Tsui, and Q. Miao, "Prognostics and health management: A review of vibration based bearing and gear health indicators," *IEEE Access*, vol. 6, pp. 665–676, 2018.
[5] S. Zhang, S. Zhang, B. Wang, and T. G. Habetler, "Deep learning algorithms for bearing fault diagnostics—A comprehensive review," *IEEE Access*, vol. 8, pp. 29857–29881, 2020.
[6] D.-T. Hoang and H.-J. Kang, "A survey on deep learning based bearing fault diagnosis," *Neurocomputing*, vol. 335, pp. 327–335, Mar. 2019.
[7] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, "Applications of machine learning to machine fault diagnosis: A review and roadmap," *Mech. Syst. Signal Process.*, vol. 138, Apr. 2020, Art. no. 106587.

[8] D. Wang, Y. Chen, C. Shen, J. Zhong, Z. Peng, and C. Li, "Fully interpretable neural network for locating resonance frequency bands for machine condition monitoring," *Mech. Syst. Signal Process.*, vol. 168, Apr. 2022, Art. no. 108673.
[9] L. Cheng, J. Lu, S. Li, R. Ding, K. Xu, and X. Li, "Fusion method and application of several source vibration fault signal spatio-temporal multi-correlation," *Appl. Sci.*, vol. 11, no. 10, p. 4318, May 2021.
[10] J. He, S. Yang, E. Papatheou, X. Xiong, H. Wan, and X. Gu, "Investigation of a multi-sensor data fusion technique for the fault diagnosis of gearboxes," *Proc. Inst. Mech. Eng. C, J. Mech. Eng. Sci.*, vol. 233, no. 13, pp. 4764–4775, Jul. 2019.
[11] V. Dave, S. Singh, and V. Vakharia, "Diagnosis of bearing faults using multi fusion signal processing techniques and mutual information," *Indian J. Eng. Mater. Sci.*, vol. 27, no. 4, pp. 878–888, 2021.
[12] H. Wang and W. Sun, "Motor bearing fault diagnosis based on wavelet packet analysis and sparse filtering," in *Proc. IEEE 4th Int. Electr. Energy Conf. (CIEEC)*, Wuhan, China, May 2021, pp. 1–7.
[13] P. Akhenia, K. Bhavsar, J. Panchal, and V. Vakharia, "Fault severity classification of ball bearing using SinGAN and deep convolutional neural network," *Proc. Inst. Mech. Eng. C, J. Mech. Eng. Sci.*, vol. 236, no. 7, pp. 3864–3877, Apr. 2022.
[14] H. Zhou, L. Cheng, L. Teng, and H. Sun, "Bearing fault diagnosis based on RF-PCA-LSTM model," in *Proc. 2nd Inf. Commun. Technol. Conf. (ICTC)*, Nanjing, China, May 2021, pp. 278–282.
[15] F. Zhou, T. Sun, X. Hu, T. Wang, and C. Wen, "A sparse denoising deep neural network for improving fault diagnosis performance," *Signal, Image Video Process.*, vol. 15, pp. 1889–1898, Jun. 2021.
[16] D. T. Hoang, X. T. Tran, M. Van, and H. J. Kang, "A deep neural network-based feature fusion for bearing fault diagnosis," *Sensors*, vol. 21, no. 1, p. 244, Jan. 2021.
[17] X. Gu and B. Zhou, "Multimodal industrial process fault detection based on LNS-DEWKECA," *Control Decis.*, vol. 35, no. 8, pp. 1879–1886, 2020.
[18] B. Zhao, X. Zhang, H. Li, and Z. Yang, "Intelligent fault diagnosis of rolling bearings based on normalized CNN considering data imbalance and variable working conditions," *Knowl.-Based Syst.*, vol. 199, Jul. 2020, Art. no. 105971.
[19] W. Zhang, C. Li, G. Peng, Y. Chen, and Z. Zhang, "A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load," *Mech. Syst. Signal Process.*, vol. 100, pp. 439–453, Feb. 2018.
[20] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Comput.*, vol. 3, no. 1, pp. 79–87, Mar. 1991.
[21] Y.-C. Bo, J.-F. Qiao, and G. Yang, "A modular neural networks ensembling method based on fuzzy decision-making," in *Proc. Int. Conf. Electr. Inf. Control Eng.*, Apr. 2011, pp. 1030–1034.
[22] Z. Geng, N. Ding, and Y. Han, "Fault diagnosis of converter based on wavelet decomposition and BP neural network," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2019, pp. 1273–1276.
[23] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 3857–3867.
[24] H. Shao, H. Jiang, F. Wang, and H. Zhao, "An enhancement deep feature fusion method for rotating machinery fault diagnosis," *Knowl.-Based Syst.*, vol. 119, pp. 200–220, Mar. 2017.
[25] H. Jiang, H. Shao, X. Chen, and J. Huang, "A feature fusion deep belief network method for intelligent fault diagnosis of rotating machinery," *J. Intell. Fuzzy Syst.*, vol. 34, no. 6, pp. 3513–3521, Jun. 2018.
[26] J. Zhu, N. Chen, and W. Peng, "Estimation of bearing remaining useful life based on multiscale convolutional neural network," *IEEE Trans. Ind. Electron.*, vol. 66, no. 4, pp. 3208–3216, Apr. 2019.
[27] F. Zhou, Z. Zhang, and D. Chen, "Bearing fault diagnosis based on DNN using multi-scale feature fusion," in *Proc. 35th Youth Academic Annu. Conf. Chin. Assoc. Autom. (YAC)*, Zhanjiang, China, Oct. 2020, pp. 150–155.
[28] Case Western Reserve University. *Bearing Data Centre*. Accessed: Dec. 2018. [Online]. Available: http://csegroups.case.edu/bearingdatacenter/home

ZHIQIANG ZHANG received the B.S. degree in automation from the Anyang Institute of Technology, in 2018, and the M.S. degree in control theory and control engineering from Henan University, in 2021. He is currently pursuing the Ph.D. degree with Shanghai Maritime University. His research interests include fault diagnosis and predictive maintenance algorithm research of electromechanical equipment.

FUNA ZHOU received the B.S. degree in basic mathematics and the M.S. degree in basic mathematics from Henan University, in 2001 and 2004, respectively, and the Ph.D. degree in power electronics and power transmission control, in 2009. She was engaged in teaching and research with the School of Computer and Information Engineering, Henan University, from 2004 to 2019. She has been engaged in teaching and research with Shanghai Maritime University, since 2019. Her research interests include fault diagnosis and predictive maintenance algorithm research of electromechanical equipment.

SIJIE LI received the B.S. degree in electrical engineering and its automation from the Taiyuan University of Technology, in 2019, and the M.S. degree in electrical engineering from Shanghai Maritime University, in 2021. Her research interests include fault diagnosis and predictive maintenance algorithm research of electromechanical equipment.